Information Processing Letters 112 (2012) 578–582

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Efficient inclusion testing for simple classes of unambiguous ω-automata

Dimitri Isaak, Christof Löding *

Informatik 7, RWTH Aachen, Germany

ARTICLE INFO

Article history:
Received 22 December 2011
Received in revised form 27 April 2012
Accepted 27 April 2012
Available online 2 May 2012
Communicated by A. Muscholl

Keywords:
Formal languages
Automata theory
ω-Automata
Unambiguous automata
Inclusion

ABSTRACT

We show that language inclusion for languages of infinite words defined by nondeterministic automata can be tested in polynomial time if the automata are unambiguous and have simple acceptance conditions, namely safety or reachability conditions. An automaton with safety condition accepts an infinite word if there is a run that never visits a forbidden state, and an automaton with reachability condition accepts an infinite word if there is a run that visits an accepting state at least once.

1. Introduction

Testing language inclusion for nondeterministic finite automata is a difficult operation from a complexity theoretic point of view. Even checking whether a nondeterministic automaton accepts all finite words is a PSPACE-complete problem (see Section 10.6 of [1]). For deterministic automata the problem is tractable because deterministic automata can be complemented and thus the inclusion problem can be reduced to the emptiness of the intersection of two automata. In [9] it has been shown that inclusion testing is possible in polynomial time if the automata are unambiguous, that is, for each input there is at most one accepting run. This result has been lifted to unambiguous automata on finite trees in [8], and can, for example, be used to derive efficient inclusion tests for certain classes of automata on unranked trees [6].

Concerning ω-automata (automata on infinite words), it is known that unambiguous Büchi automata capture the same class of ω-languages as unrestricted nondeterministic Büchi automata [2], while deterministic Büchi automata are easily seen to be less expressive. A construction directly transforming nondeterministic Büchi automata into unambiguous ones (without going through deterministic automaton models) has recently been presented in [5].

While the construction of unambiguous Büchi automata has been studied, their algorithmic properties have not yet been analyzed. In particular, the complexity of the inclusion problem for unambiguous Büchi automata is open. In [3] it is shown that inclusion testing is tractable for the class of strongly unambiguous Büchi automata. A Büchi automaton is called strongly unambiguous if it remains unambiguous even if the set of all states is declared initial. These automata are expressively complete, as shown in [4] (see also the chapter on prophetic automata in [7]).

In this paper we consider the standard notion of unambiguous Büchi automata. To obtain classes of automata with tractable inclusion problem, we look at subclasses of Büchi automata with simpler acceptance conditions. A safety automaton (sometimes called looping automaton) accepts if there is an infinite run on the input word, while a reachability automaton accepts an infinite input if there is an infinite run in which an accepting state is visited. These automata can be used to capture simple classes of properties, namely safety and guarantee properties that are frequently used in verification. We show that the inclusion problems for these two classes of unambiguous automata can be solved efficiently, basically by reducing

* Corresponding author.
E-mail address: loeding@cs.rwth-aachen.de (C. Löding).

0020-0190/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2012.04.010

them to inclusion problems for unambiguous automata on finite words.

The remainder of the paper is structured as follows. In Section 2 we introduce basic terminology and the models of automata considered in this work. In Sections 3 and 4 we show how to efficiently test inclusion for unambiguous safety and reachability automata, respectively, and in Section 5 we conclude.

2. Automata on finite and infinite words

In this section we present basic definitions and results concerning finite automata. We assume that the reader is familiar with finite automata on finite words and only briefly fix our notation.

For an alphabet Σ we denote as usual the set of finite words over Σ by Σ^* and the set of infinite words by Σ^ω. For an infinite word $\alpha \in \Sigma^\omega$ we denote the jth letter by $\alpha(j)$, i.e., $\alpha = \alpha(0)\alpha(1)\cdots$. For a nonempty finite word u we write u^ω for its infinite iteration $uuu\cdots$.

Nondeterministic finite automata (NFA) are of the form $\mathcal{A} = (Q, \Sigma, q_{\text{in}}, \Delta, F)$, where Q is a finite set of states, Σ is the input alphabet, $q_{\text{in}} \in Q$ is the initial state, $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation, and $F \subseteq Q$ is the set of accepting states. An automaton is called *complete*, if for each combination of state $q \in Q$ and letter $a \in \Sigma$ there exists a transition of the form $(q, a, q') \in \Delta$. For $q, q' \in Q$ and $u \in \Sigma^*$ we write $\mathcal{A} : q \xrightarrow{u} q'$ if there is a path in $\mathcal{A}$ from q to q' that is labeled by u. The language of finite words accepted by $\mathcal{A}$ is $L_*(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} : q_{\text{in}} \xrightarrow{w} q \in F\}$. We use the notation $L_*(\mathcal{A})$ to distinguish the language of finite words clearly from the language of infinite words accepted by $\mathcal{A}$ (as defined below).

We consider ω-automata $\mathcal{A} = (Q, \Sigma, q_{\text{in}}, \Delta, F)$ that are of the same form as NFAs. A *run* of $\mathcal{A}$ on an infinite word α is an infinite sequence $q_0 q_1 \ldots$ of states such that $q_0 = q_{\text{in}}$ and for each i the transition $(q_i, \alpha(i), q_{i+1})$ is in Δ. The run is accepting if it satisfies the *acceptance condition*, which depends on the type of the automaton. The standard acceptance condition is the Büchi condition. If $\mathcal{A}$ is considered as *Büchi automaton*, then a run is accepting if it infinitely often visits an accepting state from F. We are mainly concerned with simpler conditions, namely safety and reachability: A run of a *safety automaton* is accepting if it does not contain a state from $Q \setminus F$. A run of a *reachability automaton* is accepting if it contains a state from F.

From the above definition it is easy to see that a safety automaton can be viewed as a Büchi automaton in which all states are accepting except for possibly one rejecting sink state. In this case, a run visits F infinitely often if, and only if, it never visits the rejecting sink. Since we are working with nondeterministic automata, this rejecting sink state is not necessary and can be omitted.

Note that while each safety automaton can easily be made complete by adding a non-accepting sink, incomplete reachability automata are more expressive than complete ones, because a complete reachability automaton has no means to reject an input after a run has reached an accepting state, while an incomplete automaton can still reject if the run cannot be extended. Consider, for example, the language $(a + b)^* a^\omega$ consisting of all words with finitely many b. An incomplete reachability automaton can accept this language by looping on the non-accepting initial state on a and b, and allowing a nondeterministic transition on a to an accepting state that loops on a but has no transition for b. It is easy to see that this language cannot be accepted by a complete reachability automaton. In this paper we only consider complete reachability automata. A complete reachability automaton can be viewed as a Büchi automaton in which all states are rejecting except for one accepting sink state (looping on every input letter). Then a run visits F infinitely often if, and only if, it visits the accepting sink state. We usually denote this unique accepting state as q_{f}.

In the following, we always assume that safety and reachability automata are given in this normalized version and thus we can simply view them as special classes of Büchi automata. Using this convention, we can write $L_\omega(\mathcal{A})$ for the language of all infinite words that are accepted by $\mathcal{A}$ using the Büchi condition.

We are interested in the complexity of the inclusion problem for ω-automata, that is, the problem of deciding for two given automata $\mathcal{A}$ and $\mathcal{A}'$ whether $L_\omega(\mathcal{A}) \subseteq L_\omega(\mathcal{A}')$. A special instance of the inclusion problem is the universality problem, that is, the problem of deciding whether a given automaton accepts all words. It is well known that for NFAs these problems are PSPACE-hard (see Section 10.6 of [1]), and it is straightforward to lift this hardness result to Büchi automata, and even safety and reachability automata. Therefore, we consider a subclass of nondeterministic automata, called unambiguous.

A nondeterministic automaton $\mathcal{A}$ is called *unambiguous* if for each input there is at most one accepting run of $\mathcal{A}$ on this input. The class of unambiguous NFAs is interesting because they admit efficient algorithms but can be exponentially more succinct than deterministic automata.

Theorem 1. *(See [9].) For unambiguous NFAs the inclusion problem can be solved in polynomial time.*

The complexity of the inclusion problem (or even the universality problem) for unambiguous Büchi automata is open. The aim of this paper is to show that the inclusion problem for unambiguous safety and reachability automata can be reduced in polynomial time to the same problem for unambiguous NFAs.

3. Inclusion testing for safety automata

It is not difficult to verify that for safety automata $L_\omega(\mathcal{A}) \subseteq L_\omega(\mathcal{A}')$ holds if, and only if, $L_*(\mathcal{A}) \subseteq L_*(\mathcal{A}')$. This observation is based on the fact that an infinite word α is accepted by a safety automaton if, and only if, each of its finite prefixes is accepted by the corresponding NFA.

However, an automaton $\mathcal{A}$ can be unambiguous when viewed as an ω-automaton but ambiguous when viewed as NFA. This is illustrated in Fig. 1. The depicted safety automaton accepts all infinite words, and for each infinite word there is exactly one accepting run (the automaton always guesses the next two letters of the input). Viewed as NFA the automaton is not unambiguous. For this reason,